УДК 002.53:004.89

МЕТОД ОБЪЕДИНЕНИЯ ОНТОЛОГИЙ ПРЕДМЕТНЫХ ОБЛАСТЕЙ ЗНАНИЙ

А.Ф. Тузовский

Институт «Кибернетический центр» ТПУ
Томский научный центр СО РАН
E-mail: TuzovskyAF@kms.cctpu.edu.ru

Предложенный метод объединения онтологических моделей позволяет сформировать согласованную общую модель различных областей знаний с незначительным вмешательством специалистов. Правильность полученной онтологии проверяется с использованием методов дескриптивной логики. Рассмотрены свойства предлагаемой операции объединения онтологий.

Введение

Основной идеей онтолого-семантического подхода [1] к построению систем управления знаниями (СУЗ) является использование онтологической модели знаний организации. Данная модель знаний включает набор онтологий: онтологии верхнего уровня, определяющий основные понятия организаций в общем, онтологии среднего уровня, определяющего понятия специфичные для конкретной организации, и набора онтологий предметных областей знаний, в которых выполняется работа конкретной организации. Для разных областей знаний создаются онтологические модели для определения основных понятий, используемых в них, которые будут связываться с онтологиями более высоких уровней. Все понятия, описанные в модели, выражаются в виде формальной онтологии, которая кодируются на формальном языке, например, OWL-DL, основанном на дескриптивной логике (ДЛ).

Онтологии характеризуются концептуальной локальностью и локальностью в использовании. Системе управления знаниями организации требуется поддерживать онтологию, которая является рабочим набором большого множества различных онтологий.

Текущая используемая онтология СУЗ (модель знаний организации) является динамически развивающимся рабочим набором, состоящим из множества онтологий. Для поддержки такого набора, система управления знаниями должна динамически строить текущую онтологию, импортируя требуемые онтологии из разных подразделений организации. Общая онтология должна поддерживать логическую согласованность между отдельными онтологиями. Кроме этого, загрузка и проверка согласованности должна быть полной и насколько это возможно автоматической.

В статье рассмотрен метод объединения двух онтологий [2]. Общая онтология системы строится путем многократного добавления отдельных онтологий к общей онтологии, вместе с определениями и взаимосвязями между ними. На каждом этапе работы метода объединенная онтология проверяется на правильность с помощью дескриптивной логики.

1. Постановка задачи

Задача объединения онтологий формулируется следующим образом: даны две правильные онтологии, требуется создать третью правильную онтологию, которая представляет понятия во входных онтологиях, а также дополнительные ограничения и взаимосвязи, если они требуются. Это может быть сделано ручным просмотром двух онтологий, обнаружением синонимов, разрешением противоречий и созданием третьей онтологии. Но для СУЗ этот процесс должен быть, насколько это возможно, автоматизирован.

Для любых двух онтологий, существует много способов их объединения без создания противоречий. В общем случае, почти любое понятие в одном словаре *может быть* связано с любым понятием в другом словаре. По этой причине онтологии обычно не могут объединяться без некоторых ограничений и подсказок со стороны проектировщика и/или администратора системы.

Описываемый метод требует небольшого набора естественных ограничений на организацию онтологий, вместе с набором простых подсказок от разработчика онтологии или администратора системы. При наличии двух онтологий, которые следуют этим правилам, метод автоматически создает полную объединенную онтологию, распространяя результаты сделанных подсказок. Метод объединения находит конфликты между определениями, тем самым гарантируя, что составная онтология будет или правильной, или будет отклонена в связи с логической несогласованностью.

Онтология естественно интерпретируется, как граф классов. Каждое понятие является вершиной, а отношения включения (отношения класс-подкласс) являются дугами. Класс без базового класса является вершиной графа; класс может иметь любое количество базовых классов (т. е. поддерживается множественное наследование). Онтология может использовать (т. е. *импортировать*) элементы из других онтологий в свои определения. Когда имеются импортированные понятия, то данный метод применяется рекурсивно, к каждой импортированной онтологии. Для простоты, в данном подходе, предполагается, что онтология не имеет ссылок на внешние онтологии. Этого можно добиться, полностью расширяя все импортируемые понятия в онтологии до использования данного метода. Без потери общности, далее будут рассматриваться онтологии, являющиеся единым графом понятий с одной корневой вершиной.

Базовым ограничением на онтологии является то, что они должны быть организованы в виде леса корневых графов или деревьев. По существу это запрещает графы с циклами, т. е. вершины, которые являются родителями одного из своих собственных предков. Ограничение онтологий деревьями позволяет манипулировать онтологиями целиком с использованием операций на небольших наборах корневых вершин. Предлагаемые ограничения являются естественными, в особенности, когда онтология проектируется как расширение онтологии более высокого уровня.

Объединенная онтология проверяется с помощью алгоритма ДЛ, который используется для проверки отдельных онтологий, как это поясняется в [3].

В дополнение к иерархиям связанных понятий, также могут быть и логические взаимосвязи между любыми понятиями в онтологиях. Например, два понятия в разных онтологиях могут быть синонимами, или не корневой класс может быть подклассом некоторого класса в другой онтологии.

Иногда, также может быть важным явно заявить, что два понятия являются, по определению, различными (непересекающимися). Это объявление может быть необходимо, так как два схожих, но различных понятий могут не иметь каких-либо отличительных характеристик в определении онтологии. Эти отношения могут быть явно заявлены в виде утверждений (называемых аксиомами в логическом программировании), которые могут быть включены в качестве подсказок при объединении онтологий (утверждение *disjoint* (c_i, c_j) дескриптивной логики). Аксиомы добавляются в виде утверждений базы знаний модели ДЛ, когда проверяется правильность объединенной онтологии.

Концептуально, аксиомы представляют собой глобальные ограничения, но они описываются утверждениями языка онтологий, которые обрабатываются точно так же, как определения понятий и отношений. Аксиомы транслируются в размеченные ребра графа онтологии, а затем в соответствующие утверждения логики.

Аксиомы могут определять отношения между понятиями с одной онтологии или между понятиями в разных онтологиях. В первом случае, они могут быть включены в онтологию (как дополнительные отношения). Во втором случае, они могут быть в отдельной онтологии, которая будет импортировать онтологии, на которые ссылаются аксиомы. Аксиомы предоставляют способ явно управлять и даже переопределять используемый по умолчанию метод объединения графов. Этот подход должен использоваться, когда имеется специфический домен знаний, который определяет синонимы межонтологических отношений.

2. Метод объединения

Вначале введем простое определение обычной онтологии. *Онтология* это упорядоченная пара, $O=(C, R, P_c)$, где C – набор вершин, R – набор отношений, а P_c – множество ассиметричных, транзитивных, нерефлексивных бинарных отношений, являющихся отношением частичного порядка на множестве понятий C, которые определяют связи между понятиями C и отношениями R.

Далее для пояснения метода будет использоваться более простое определение онтологии, основанное на графовой модели: *Онтология* это упорядоченная пара, $O=(C, R)$, где C – набор вершин, R – набор ребер. Каждое понятие в онтологии представлен в виде вершины в C. Направленное ребро описывает включение понятий или бинарное отношение между понятиями. *Корневая вершина* не имеет входящих дуг. Вершины и дуги онтологий O_1 и O_2 имеют уникальные имена. Под *правильной онтологией* понимается такая онтология, относительно которой будет выполняться доказательство ее согласованности, например, используя ДЛ, как это показано в [3].

Целью предлагаемого метода является создание новой онтологии, которая представляет собой объединение исходных онтологий, включая отношения между понятиями в двух онтологиях. Входными данными являются две правильные онтологии O_1, O_2: $O_1=(C_1, R_1)$ и $O_1=(C_2, R_2)$, а также набор подсказок (связей): множество L упорядоченных пар (c_1, c_2), где $c_1 \in C_1$ и $c_2 \in C_2$. Требуется, чтобы новая онтология O_{1+2} была правильной, со всеми понятиями онтологий O_1 и O_2, а также с отношениями, заявленными в подсказках (*связях*). Если результирующая онтология содержит противоречие, тогда результат метода является неуспешным (ошибочным).

Метод объединения состоит из 3 шагов: слияние, связывание конкретных корневых вершин, и, затем, проверка правильности новой онтологии, что в ней не появились противоречия.

Шаг 1: Слияние O_1 и O_2 для формирования O_{1+2}. По определению: $O_{1+2}=O_1 \cup O_2$, где $O_1 \cup O_2=(C_1 \cup C_2, R_1 \cup R_2)$. Объединение всегда может быть вычислено, но результат в принципе может содержать противоречия. При желании, $O_1 \cup O_2$ онтология может быть проверена на правильность на данном шаге.

Шаг 2: Связывание взаимосвязанных иерархий, описываемое набором *связей*. Для каждой пары $(c_1,c_2) \in L$ в O_{1+2} заносится утверждение *subClassOf*(c_1,c_2), т. е. создается новая дуга в графе. После этого шага, составная онтология определяется следующим образом: $O_{1+2}=(C_1 \cup C_2,\ R_1 \cup R_2 \cup L)$. При предположении, что элементы этих связей являются правильными (т. е. каждая $c_1 \in C_1$ и $c_2 \in C_2$), эта операция всегда может быть выполнена, но возможно, что использование связей создаст противоречие. Опять, на этом шаге может быть выполнена проверка правильности онтологии $O_1 \cup O_2$.

Шаг 3. Проверка правильности составленной онтологии O_{1+2}. Если онтология O_{1+2} является правильной то тогда ***УСПЕХ*** (цель достигнута), иначе ***НЕУДАЧА***. Алгоритм завершен.

При объединении двух онтологий, могут появляться дополнительные логические ограничения или взаимосвязи, которые применяются к объединенной онтологии. В частности, иногда требуется определить взаимосвязи между понятиями в двух исходных онтологиях. Эти взаимосвязи должны быть добавлены к онтологии как дополнительные ребра. Это выполняется путем создания небольшой, фрагментарной онтологии, которая заявляет аксиомы, которые добавляется к O_{1+2}, используя выше описанный метод. Общий процесс объединения может включать объединение нескольких онтологий, для построения конечной онтологии.

3. Свойства операций объединения онтологий

Метод объединения определяет операцию в универсуме правильных онтологий. Эта операция имеет некоторые простые свойства, которые следуют из определений и монотонности ДЛ [4].

3.1. Операция объединения онтологий не является замкнутой

Объединение онтологий не является замкнутым над набором правильных онтологий. Если $valid(O_i)$ обозначает, что онтология O_i является правильной, то $valid(O_1) \wedge valid(O_2)$ не предполагает $valid(O_{1+2})$. Это является верным, так как O_{1+2} может иметь взаимосвязи между O_1 и O_2, которые создают противоречия, которые не применимы к отдельным онтологиям O_1 или O_2. Для иллюстрации, как это может возникнуть, рассмотрим пример из [5] объединения двух простых онтологий (рис. 1).

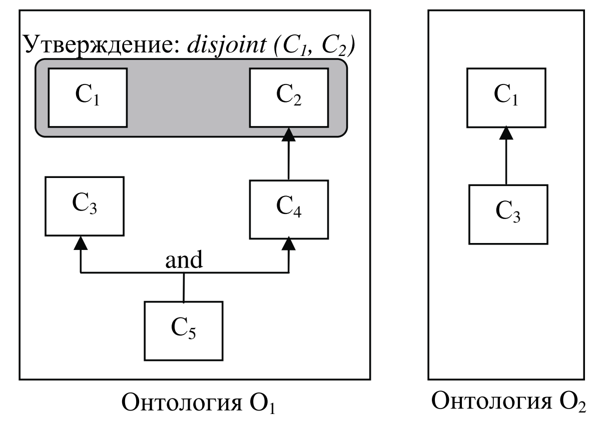


Рис. 1. *Две простые онтологии, каждая из которых является правильной*

Онтология O_1 описывает 5 понятий и взаимосвязи между ними. Понятие C_4 определено как производное от понятия C_2, а C_5 определено, как производное от C_3 и C_4 (множественное наследие). Кроме этого определено ограничение на понятия C_1 и C_2: C_1 не может быть C_2. Это ограничение означает, что объект может быть или C_1 или C_2, но не совместно C_1 и C_2. Онтология O_1 является правильной, т. е. не содержит противоречий.

Онтология O_2 определяет взаимосвязи между понятиями C_1 и C_3, которые являются теми же самыми понятиями, как C_1 и C_3 в онтологии O_1. В онтологии O_2 понятие C_3 является производным от понятий C_1, т. е. понятие C_3 является понятием C_1. Онтология O_2 является правильной онтологией.

Когда эти две онтологии объединяются, они формируют третью онтологию, O_{1+2} (рис. 2). Обе исходные онтологии определяют взаимосвязи между понятиями C_1 и C_3, поэтому онтология O_{1+2} включает все эти взаимосвязи.

При логическом анализе онтологии O_{1+2} обнаруживается противоречие. Учитывая, что C_1 и C_2 определены как непересекающиеся, из отношений вхождения (наследования) следует, что C_3 и C_4 также должны быть не пересекающимися. Понятие C_5 было определено как производным от C_3 и C_4, но это не возможно выполнить, так как задано утверждение $disjoint(C_3, C_4)$. В связи с этим, объединенная онтология является не правильной.

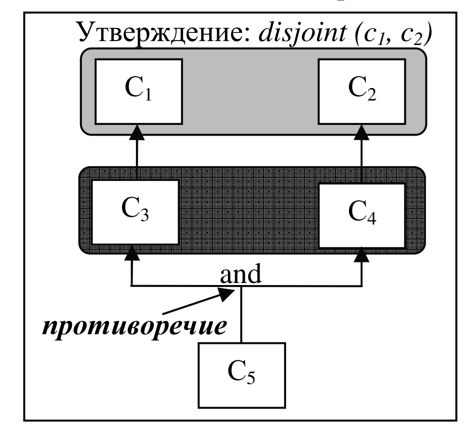


Рис. 2. *Появление противоречия при объединении правильных онтологий*

Этот пример показывает, что объединение двух правильных онтологий не обязательно дает правильную онтологию, т. е. операция объединения не является замкнутой. Противоречие возникает из-за того, что *исходные онтологии имеют противоречивые утверждения, по крайней мере, для одного понятия*. Логика обнаруживает противоречие, даже в случае, когда ни одна из исходных онтологий не содержит противоречивые утверждения, содержащие понятие C_5.

3.2. Операция объединения является коммуникативной

Хотя две составные онтологии не всегда являются правильными, но когда сумма отношений O_1+O_2 является правильной, то сумма O_2+O_1 также должна быть *правильной*, т. е. $valid(O_{1+2}) \Leftrightarrow valid(O_{2+1})$. Интуитивно понятно, что O_{1+2} является тем же самым набором, что и O_{2+1}, поэтому они или оба правильные или оба неправильные. Формально это следует из определений и монотонности ДЛ.

Рассмотрим две онтологии, $O_1=(C_1, R_1)$ и $O_2=(C_2, R_2)$ и набор связей L. Операция объединения *compose* (O_1, O_2, L) создает онтологию O_{1+2}, которая определяется как: $O_{1+2}=(C_1\cup C_2, R_1\cup R_2\cup L)$. Аналогично, *compose* (O_2, O_1, L) создает онтологию O_{2+1}, которая определяется как: $O_{2+1}=(C_2\cup C_1, R_2\cup R_1\cup L)$. Эти два графа будут создавать один и тот же набор утверждений в ДЛ, возможно только за исключением порядка их следования. Из того факта, что ДЛ является монотонной, следует, что вывод из базы знаний $БЗ_{1+2}$ будет правильным и для базы знаний $БЗ_{2+1}$. Поэтому, $satisfiable(БЗ_{1+2})\Leftrightarrow satisfiable(БЗ_{2+1})$ и из определения в [3] можно записать, что $valid(O_{1+2})\Leftrightarrow valid(O_{2+1})$.

3.3. Операция объединения является ассоциативной

Операция объединения также имеет ассоциативные свойства, т. е. $valid((O_{1+2})+O_3)\Leftrightarrow valid(O_1+(O_{2+3}))$. Это свойство следует из доказательства, аналогичного приведенному выше. Рассмотрим три онтологии $O_1=(C_1, R_1)$, $O_2=(C_2, R_2)$, $O_3=(C_3, R_3)$ вместе со связями L_{1+2}, L_{2+3}, L_{1+3}. Три онтологии объединяются за два шага. Для построения $O_{(1+2)+3}$, на первом шаге объединяем 1 и 2 онтологии: $O_{1+2}=$ *compose* $(O_1, O_2, L_{1+2})=(C_1\cup C_2, R_1\cup R_2\cup L_{1+2})$.

Предположим, что утверждение $valid(O_{1+2})$ является *истинным*, т. е., $satisfiable(БЗ_{1+2})$ является ***истинным***. На втором шаге добавляет онтологию O_3: $O_{(1+2)+3}=$*compose* $(O_{1+2}, O_3, (L_{1+3}\cup L_{2+3}))=((C_1\cup C_2)\cup C_3, (R_1\cup R_2\cup L_{1+2})\cup R_3\cup L_{1+3}\cup L_{2+3})=((C_1\cup C_2\cup C_3), (R_1\cup R_2\cup R_3\cup L_{1+2}\cup L_{1+3}\cup L_{2+3})$.

Предположим, что утверждение $valid(O_{(1+2)+3})$ является *истинным*. Для построения $O_{1+(2+3)}$ можно выполнить два аналогичных объединения: $O_{2+3}=$*compose* $(O_2, O_3, L_{2+3})=(C_2\cup C_3, R_2\cup R_3\cup L_{2+3})$ и $O_{1+(2+3)}=$*compose* $(O_1, O_{2+3}, (L_{1+3}\cup L_{2+3}))=((C_2\cup C_3)\cup C_1, (R_2\cup R_3\cup L_{2+3})\cup R_1\cup L_{1+3}\cup L_{1+2})=((C_1\cup C_2\cup C_3), (R_1\cup R_2\cup R_3\cup L_{1+2}\cup L_{1+3}\cup L_{2+3})$.

Это такой же набор, как и $O_{(1+2)+3}$, за исключением порядка элементов. В связи с этим утверждение $valid(O_{1+(2+3)})$ является *истинным*. Из определения правильности и монотонности ДЛ следует, что $satisfiable(БЗ_{(1+2)+3})\Leftrightarrow satisfiable(БЗ_{1+(2+3)})$ и поэтому $valid(O_{(1+2)+3})\Leftrightarrow valid(O_{1+(2+3)})$. Это свойство предполагает, что для любой O_{1+2+3}, полученной, как описано выше, является верным: $valid(O_{1+2+3})\Rightarrow valid(O_{1+2})\wedge valid(O_{1+3})\wedge valid(O_{2+3})\wedge valid(O_1)\wedge valid(O_2)\wedge valid(O_3)$.

3.4. Операция объединения не является транзитивной

Можно увидеть, что из свойства не замкнутости операция объединения не является транзитивной. Т. е. из правильности $valid(O_{1+2})\wedge valid(O_{2+3})$ не следует правильность $valid(O_{1+3})$. Фактически, даже если верно, что $valid(O_1)\wedge valid(O_2)\wedge valid(O_3)\wedge valid(O_{1+2})\wedge valid(O_{2+3})$, то все еще возможно, что O_{1+3} будет иметь противоречия, как это было рассмотрено выше. Аналогично, $valid(O_{1+2})\wedge valid(O_{2+3})$ не предполагает, что $valid(O_{1+2+3})$ является верной.

Заключение

Рассмотренный метод объединения двух онтологий позволяет автоматизировать формирование общей онтологической модели и поддерживать логическую согласованность общей объединенной онтологии, что является основой для работы с единой моделью знаний организации. Прикладные подсистемы системы управления знаниями могут использовать итоговую онтологию в рамках семантического Web-портала для выполнения разнообразных задач по управлению знаниями – категоризации, рекомендации, поиска.

СПИСОК ЛИТЕРАТУРЫ

1. Тузовский А.Ф. Онтолого-семантический подход к созданию систем управления знаниями организаций // Интеллектуальные системы (INTELS'2006): Труды 7 Междунар. симп. – Краснодар, 2006. – С. 286–290.
7. McGrath R.E. Semantic infrastructure for a ubiqutious computing environment [Электронный ресурс]. – 2005. – Режим доступа: http://www.cs.uiuc.edu/research/techreports.php?report=UIUCDCS-R-2005-2587&downioad=pdf
3. Тузовский А.Ф. Работа с онтологической моделью организации на основе дескриптивной логики // Известия Томского политехнического университета. – 2006. – Т. 309. – № 7. – С. 134–137.
4. The Description Logic handbook: theory, implementation, applications // Ed. F. Baader. – Cambridge: Cambridge University Press, 2003. – 564 p.
5. Anand R., McGrath R., Campbell R., Mickunas M. Use of Ontologies in a Pervasive Computing Environment // Knowledge Engineering Review. – 2004. – V. 18. – № 3. – P. 209–220.